YF-TFT320240-3.5

嵌入式液晶显示器

用说明书

郑州远峰电子

本说明书仅适用于郑州远峰电子定做生产的 YF-TFT320240-3.5 嵌入式彩色液晶显示器。如果在您对本说明书或产品有任何疑问，欢迎拨打电话咨询。

目　　录

1　简介

YF-TFT320240-3.5 是郑州远峰电子专门针对单片机用户而定制设计的液晶显示器(带触摸屏)，采用原装 NEC3.5 英寸、分辨率为 320x240 的真彩 TFT 屏，提供一个简单的高速 8 位总线与单片机连接，支持 256 色。可以直接与 MCS51、MCS96、MC68、ARM 以及 DSP 相连。直接输入 X、Y 坐标，无须计算地址。

低功耗、轻薄设计（高度 8.5mm）、宽温（-20 度到 70 度）、亮度可调节（软件调节 8 种亮度）、低功耗模式（软件关断显示）。

适合各种仪器仪表、工业设备的应用，其低功耗、轻薄设计亦能满足单节锂电池供电设备的需求。

软件性能全面升级，提供以下功能：

1、快速清屏功能；只需发送一条指令，控制板在 16.6 毫秒内以指定的颜色对整个画面进行清屏，清屏过程无须单片机的干预，极大地提高了开机和单一背景色的显示速度。

2、提供 8 点写模式；使得彩色屏的操作与单色屏一样，极大地提高了汉字、英文字母、数字、以及单色位图的显示速度。

3、提供灵活的地址自动加一功能；地址自动加一的方向可以任意设置为 X 方向或 Y 方向。地址沿 X 方向自动加一时，遇到行尾将自动跳到下一行的行首。地址沿 Y 方向自动加一时，遇到列尾将自动跳到下一列的列首。

通过以上各种加强的功能，使得普通的单片机驱动彩色屏，也可以得到非常流畅的显示效果。

2 工作原理与系统框图

该显示器接口采用并行总线方式（数据总线 D[7：0]、地址总线 A[1：0]、片选/CS、读/RD、写/WR），可以很方便地连接到单片机或微处理器的总线上，用户可以象使用普通存储器一样使用它。由于采用了命令方式，只需要两个地址信号线，大量节约了处理器的存储空间，减少了接口信号的个数，有效简化了硬

件系统连接。

YF-TFT320240-3.5 的显示存储器同液晶屏幕上的像素点一一对应，用户可以把这个存储器理解为“显示映像”存储器。如果希望在液晶屏幕上某一位置显示文字或图形，只需要向存储器内对应区域写入相应的数据即可。

为提高读写速度，简化程序，显示屏中每个点影射显示存储器中的一个字节，显示屏上的 X、Y 坐标与显示存储器的地址一一对应。因此，只需输入 X、Y 坐标便可直接读写相应点数据，不用计算像素点在显示存储器中的地址，写入数据后 X 坐标自动加 1，写满一行后自动换行，用户也可实现 Y 坐标自动加 1。

显示存储器的一个字节由 8 位构成，显示器屏幕上的一个“像素”点由 R、G、B 三个“点”来组成。一个字节对应一个像素，位分配原则是是 R3-G3-B2，也就是红色占高三位，绿色占中间三位，蓝色为低 2 位。因此，共可显示 8 种红色，8 种绿色，4 种蓝色。通过红、绿、蓝的混合，总共可以得到 256 种颜色。

YF-TFT320240-3.5 的基本原理框图如图 1 所示。DC/DC 电压变换器产生液晶显示所需要的各种驱动电压，背光驱动电路则产生 LED 背光灯所需要的供电电源。逻辑控制电路采用 CPLD 器件，自主研制的优化逻辑，利用独有的缓冲技术，让显示与写入数据同时进行，完全杜绝雪花现象，实现了画面的高速更新，且互不干扰。

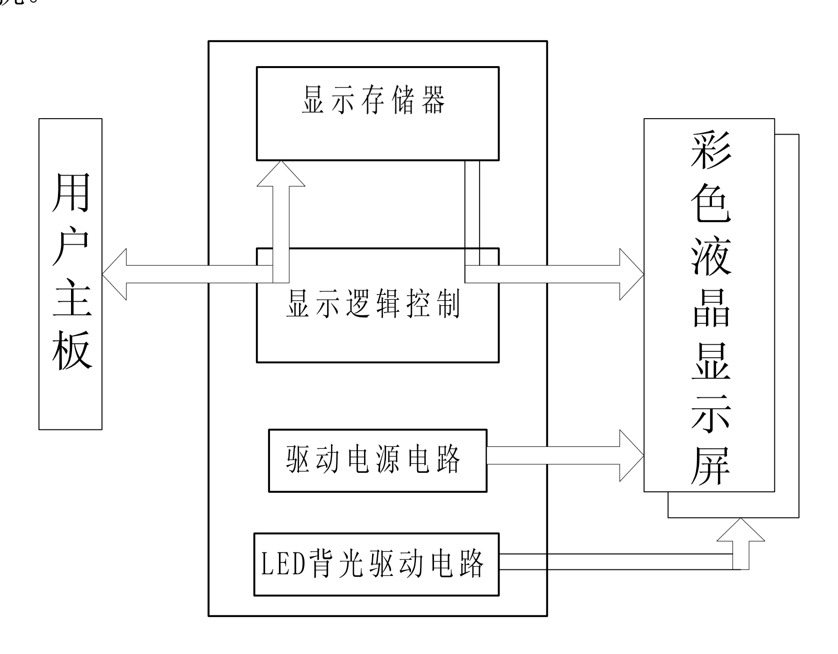


图 1 YF-TFT320240-3.5 原理框图

3 性能参数

图像点阵	320×RGB×240 行
视域尺寸	70.08×52.56mm
视域对角线	3.5 英寸
接口方式	总线方式
显示颜色	TFT 256 色
背光类型	LED
背光亮度	250
工作电压	3.3V/5V
消耗功率	0.7W
外型尺寸	95(77.8)×64.5×8.5mm
工作环境温度	-30～85℃
保存温度	-40～95℃

4 引脚定义

表一 引脚定义 (插座类型：1mm 的 24 脚 FFC)：

引脚	符号	功能
1	GND	
2	GND	
3	VCC1	YF-TFT320240-3.5A　+3.3V
		YF-TFT320240-3.5B　+5V
4	/RD	读操作信号，低电平有效。
5	/WR	写操作信号，低电平有效。
6	/CS	片选信号，低电平有效
7	A0	地址
8	A1	地址
9	DATA0	数据总线
10	DATA1	数据总线
11	DATA2	数据总线
12	DATA3	数据总线
13	DATA4	数据总线
14	DATA5	数据总线
15	DATA6	数据总线
16	DATA7	数据总线

17	XR	触摸屏-右
18	YD	触摸屏-下
19	XL	触摸屏-左
20	YU	触摸屏-上
21	VCC2	YF-TFT320240-3.5A　+3.3V
22		YF-TFT320240-3.5B　+5V
23	N. C.	
24	N. C.	

注意：

1．控制板分 YF-TFT320240-3.5A 和 YF-TFT320240-3.5B 两种型号，主要差异是 VCC 的供电电压。未做特殊说明下，提供的都是 B 型，5V 供电。

2．不论 A 型还是 B 型，所有接口信号（/CS、/WR、/RD、A[1：0]、D[7：0]）都是兼容 3.3V 和 5V 逻辑电平的，可以与 5V 系统或 3.3V 系统直接相连接，不需要任何电平转换电路。

3．触摸屏为可选配件。

5 显示存储器与像素对应关系（256 色）

列坐标（X）是以字节为单位的，一字节包含 8 位，对应一个象素；因此，列坐标 X 取值范围是 0-319。行坐标（Y）取值范围是 0-239。

像素格式为 R3G3B2。也就是红色占高三位，绿色占中间三位，蓝色为低2位。因此，共可显示 8 种红色，8 种绿色，4 种蓝色。通过红、绿、蓝的混合，总共可以得到 256 种颜色。

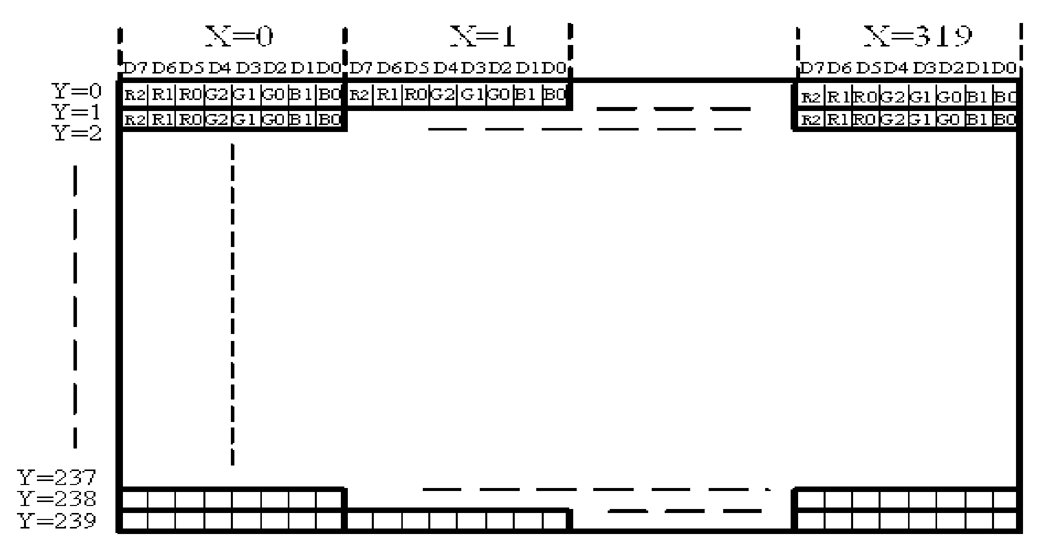


图 2　显示存储器与像素对应关系

6 颜色的组成

表 2 颜色组成

字节数据		D7、D6、D5	D4、D3、D2	D1、D0
	颜色灰度	R2、R1、R0	G2、G1、G0	B1、B0
基本颜色	最黑	000	000	00
	亮蓝	000	000	11
	亮绿	000	111	00
	亮青	000	111	11
	亮红	111	000	00
	亮紫	111	000	11
	亮黄	111	111	00
	亮白	111	111	11
蓝色灰度	最黑	000	000	00
	较暗	000	000	01
	较亮	000	000	10
	最亮	000	000	11
绿色灰度	最黑	000	000	00
	较暗	000	001	00
	…	…	…	…
	较亮	000	110	00
	最亮	000	111	00
红色灰度	最黑	000	000	00
	较暗	001	000	00
	…	…	…	…
	较亮	110	000	00
	最亮	111	000	00

7 寄存器描述（基本功能-单点写）

共有 4 个寄存器，分别为列地址、行地址、状态控制寄存器、显示数据。

/CS	A1A0	/WR	功能
0	00	0	列地址寄存器 X
0	01	0	行地址寄存器 Y
0	10	0	控制寄存器 CMD
0	11	0	数据寄存器 DAT

列地址寄存器（X）：由于列地址取值范围是从 0-319，占 9bit，所以列地

址寄存器（X）必须连续写两次，第一次写低 8 位，第二次写高 1 位。在两次写操作之间不可以对其它三个单元做任何操作。

第一次：

X7	X6	X5	X4	X3	X2	X1	X0

第二次：

-	-	-	-	-	-		X8

行地址寄存器（Y）：与列地址寄存器（X）相似，由于行地址取值范围是从 0-239，占 8bit，所以行地址寄存器（Y）只需要写一次。

Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

控制寄存器：在实现基本功能（单点写）时，不需要使用控制寄存器，直接将该寄存器写 0 就可以了。在基本功能下，一个像素对应一个字节，如上节所描述的颜色表。

0	0	0	0	0	0	0	0

数据寄存器 DAT：显示数据通过该寄存器写入和读出，每次读写操作后地址自动沿 X 方向加一。一次读写一个像素。

D7	D6	D5	D4	D3	D2	D1	D0

8 显示数据读写方式

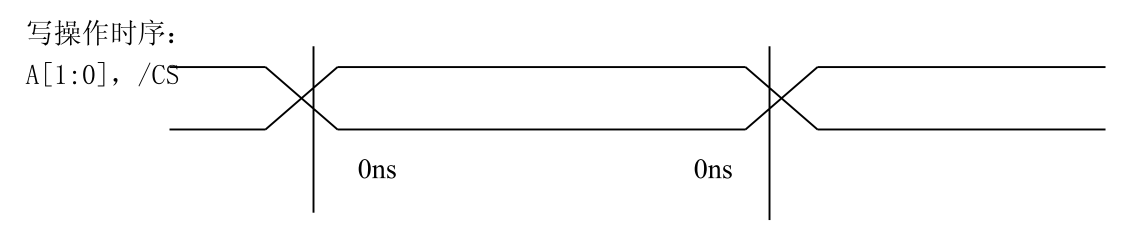
首先必须指定行地址 Y，以及列地址 X。然后就可以将该行从地址 X 开始的数据连续进行读写操作，无须重新设置 X 和 Y。

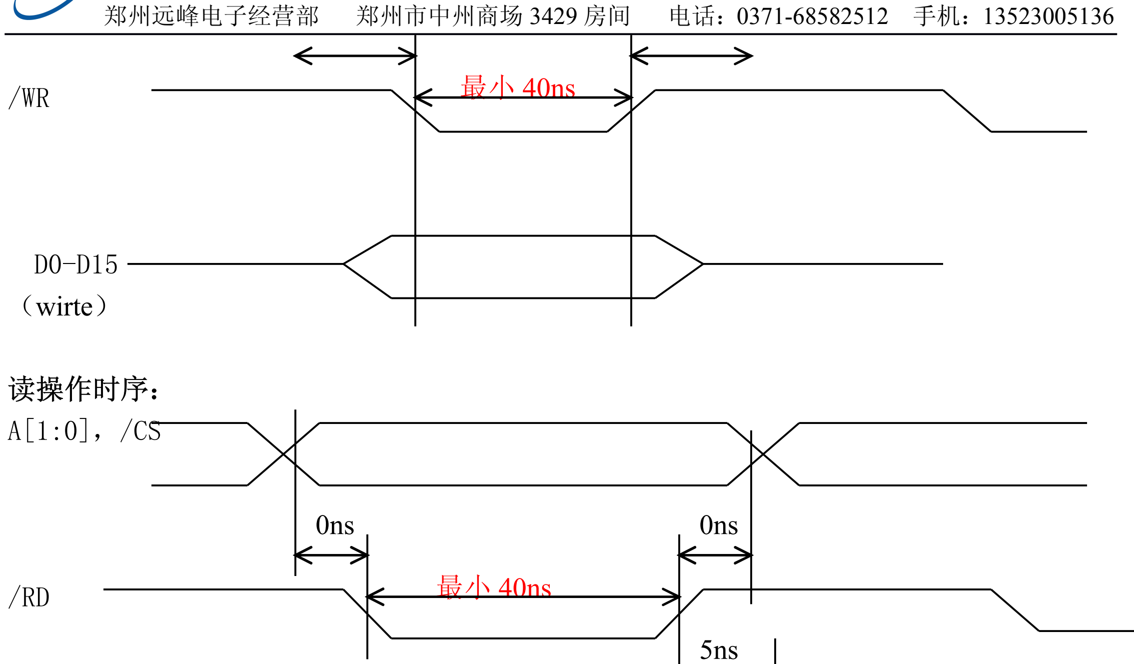
在显示数据的每次读写操作后，列地址 X 将自动加 1。当地址加到行尾时，地址将跳到下一行的行首。

当要读写一个新的行时，必须重新设置 X、Y。

9 总线时序

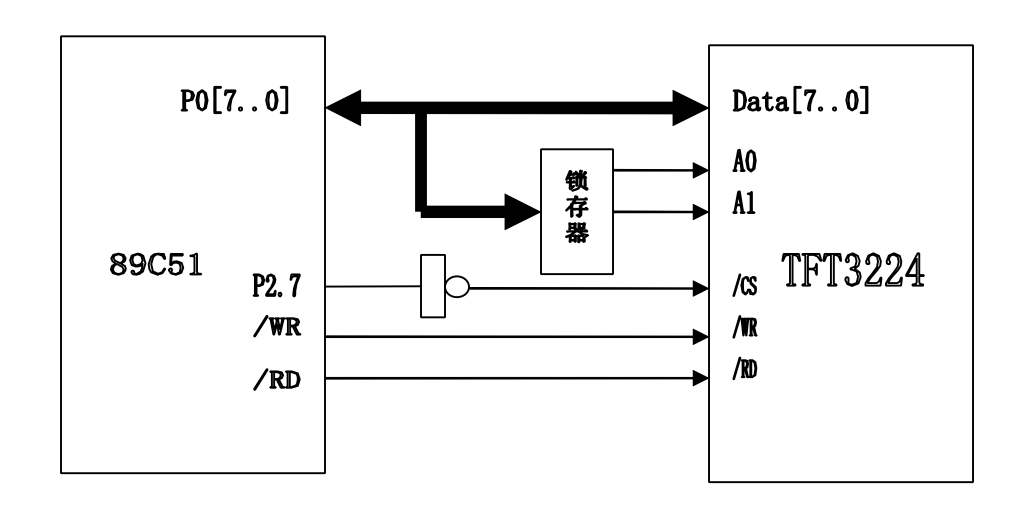
写操作时序：



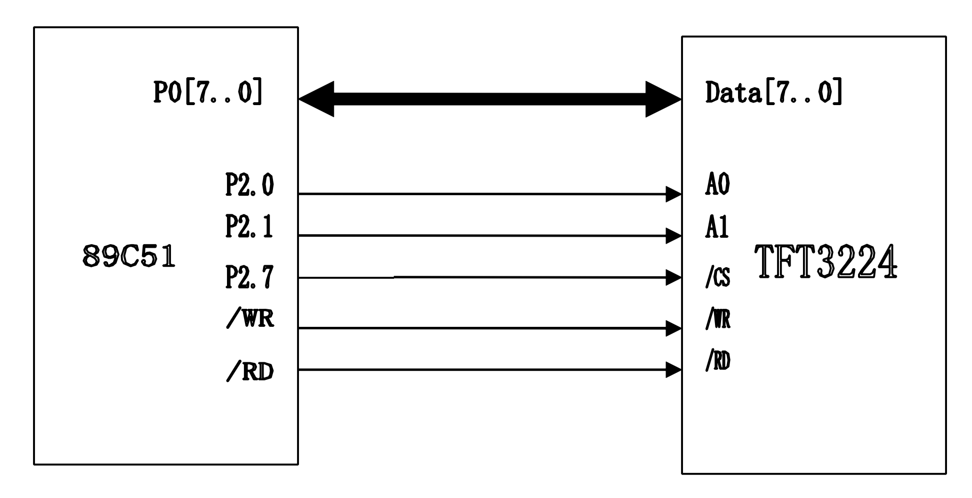


10 接口电路（以 MCS51 单片机为例）

典型接口电路：



DEMO 板接口电路：（省去了地址锁存器和译码器）

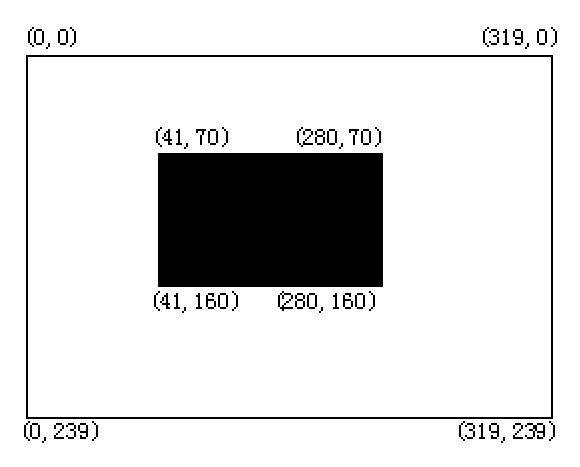


以上两种接口电路的端口地址分别为：

寄存器名	端口地址（典型）	端口地址（DEMO）
列地址寄存器	8000H	0000H
行地址寄存器	8001H	0100H
控制寄存器	8002H	0200H
读写显示数据	8003H	0300H

11 软件编写

图片数据的显示：



将图中间区域填充成蓝色。
采用行操作模式。
在每行的写操作前，先设置 X、Y。

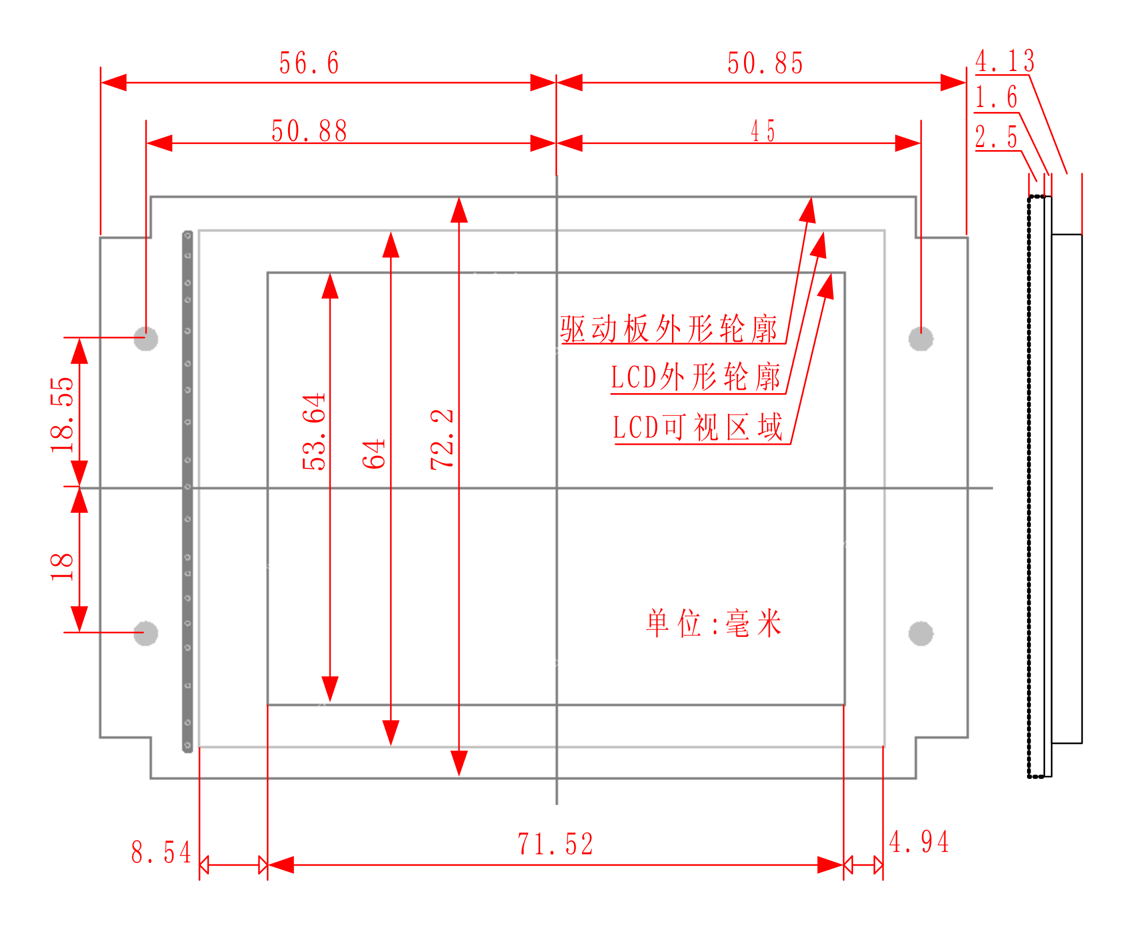
```
#include <reg51.h>
#include <absacc.h>
#define X_ADDR XBYTE[0x0000]
```

```
#define Y_ADDR XBYTE[0x0100]
#define CMD XBYTE[0x0200]
#define DAT XBYTE[0x0300]

main()
{
unsigned int x;
unsigned char y;
CMD=0x00;
for ( y=70; y<=160; y++ )
        {
        X_ADDR = 41;
        X_ADDR = 0;//X[8]=0
        Y_ADDR = y;
        for( x= 41; x<=280; x++){ DAT = 0x03;}
        }

}
```

12 机械尺寸与布局



13 提高功能

真彩色 LCD 控制板性能全面升级，提供以下提高功能：

1、快速清屏功能；只需发送一条指令，控制板在 16.6 毫秒内以指定的颜色对整个画面进行清屏，清屏过程无须单片机的干预，极大地提高了开机和单一背景色的显示速度。

2、提供 8 点写模式；使得彩色屏的操作与单色屏一样，极大地提高了汉字、英文字母、数字、以及单色位图的显示速度。

3、提供灵活的地址自动加一功能；地址自动加一的方向可以任意设置为 X 方向或 Y 方向。地址沿 X 方向自动加一时，遇到行尾将自动跳到下一行的行首。地址沿 Y 方向自动加一时，遇到列尾将自动跳到下一列的列首。

通过以上各种加强的功能，使得普通的单片机驱彩色屏，也可以得到非常流畅的显示效果。

提高功能是通过控制寄存器和显示数据寄存器的复用来实现的。

控制寄存器：

-	Back_disable	Power_Off	Inc_dir	Clear_en	Mode[1]	Mode[0]	Dot8_en

Back_disable：控制寄存器 bit[6]，初始值为 0；

该位是背景色禁止位，只在 8 点写模式下有用。Back_disable 为 1，禁止背景色写入；为 0 则允许背景色写入。

初期使用者，可以不考虑该位，将其置为 0 就可以了。

详细用法参见 Dot8_en。

Power_Off：控制寄存器 bit[5]，初始值为 0；

低功耗模式控制位。为 0 正常工作。为 1 进入低功耗模式，关断显示并保存画面，当重新回到正常工作模式下时，画面保持不变。

Inc_dir：控制寄存器 bit[4]，初始值为 0；

设定地址自动加一的方向，为 0 沿 X 方向自动加一，为 1 沿 Y 方向自动加一。地址沿 X 方向自动加一时，遇到行尾将自动跳到下一行的行首。地址沿 Y 方向自动加一时，遇到列尾将自动跳到下一列的列首。

Clear_en：控制寄存器 bit[3]，初始值为 0；

清屏使能位。该位为 1 时，启动清屏操作，控制板将自动按照定义的背景色颜色（见 bit[2]）填充整个画面，该过程需要耗时 16.6 毫秒。在填充过程中，无须单片机的干预。单片机使能该位后，等待 16.6 毫秒，再将该为写为 0，重新回到正常模式工作。可见，在进行清屏操作前，必须先设置背景色颜色。

Mode[1：0]：控制寄存器 bit[2：1]，初始值为 00；

数据寄存器 DAT（A1A0==11）的功能定义。

Mode = 00 ：数据寄存器 DAT 是像素数据写入寄存器。

在单点模式下，一次写入 1 个像素，数据格式是 R3G3B2；在 8 点模式下，一次写入 8 个像素，数据格式是每一位对应一个像素。

Mode = 01 ：数据寄存器 DAT 是前景色颜色写入寄存器。

前景色颜色用于 8 点写入模式。数据格式是 R3G3B2。

Mode = 10 ：数据寄存器 DAT 是背景色颜色写入寄存器。

背景色颜色用于清屏和 8 点写入模式。数据格式是 R3G3B2。

Mode = 11 ：数据寄存器 DAT 是亮度值写入寄存器。

最低 3 位为有效位 Brightness[2:0]，定义 8 种亮度。第 7 级亮度最强，第 0 级亮度最低。在不同的亮度下，VCC1 的电流是不同的。

-	-	-	-	-	Bright[2]	Bright[1]	Bright[0]

Dot8_en：控制寄存器 bit[0]，初始值为 0；

8 点写模式使能位。该位为 1 使能 8 点写模式，为 0 则是前面介绍的单点写的基本功能。读操作只能是单点操作模式。

在 8 点写模式下，一次写入 8 个像素点（A1A0=11），一个像素点对应数据寄存器的一位。

在 8 点写模式下：

若像素对应位为 1 则将前景色颜色值赋给对应像素；

若像素对应位为 0 且 Back_disable 为 0 则将背景色颜色值赋给对应像素；

若像素对应位为 0 且 Back_disable 为 1 则对应像素保持不变。

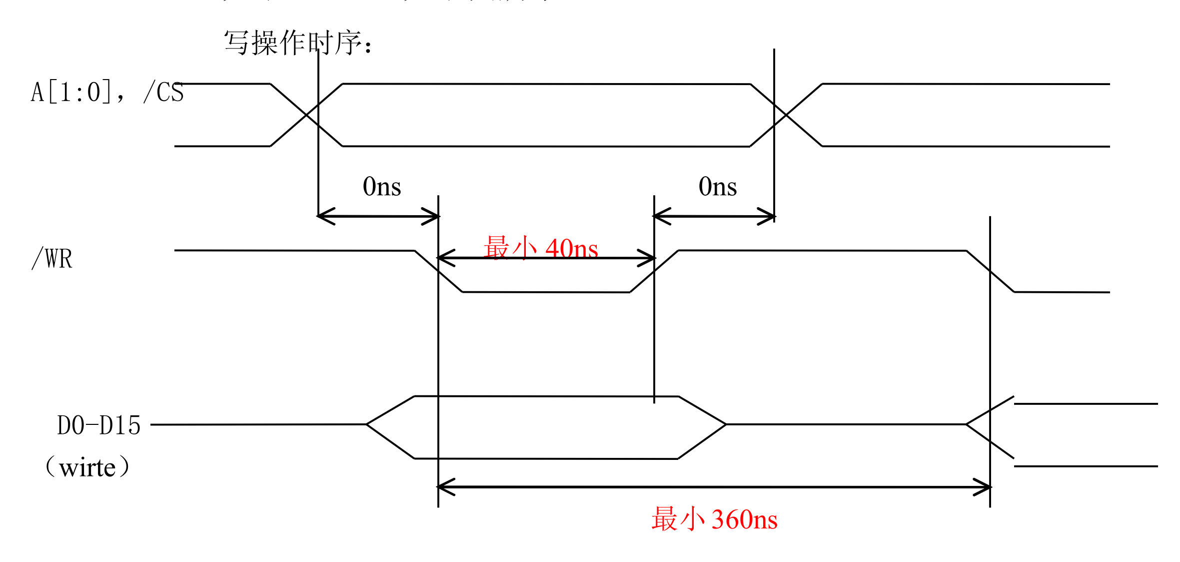
可见，在进行 8 点写操作前，必须先设置背景色颜色和前景色颜色。

背景色禁止位 Back_disable 提供了只修改前景色的功能，提高了应用的灵活性。

在 8 点写模式下，地址将自动加 8。

在 8 点写模式下，连续写数据寄存器（A1A0=11）的时间间隔必须大于 360ns，如下图所示。

写操作时序：



提高功能的例程：

```
#define X_ADDR XBYTE[0x0000]
#define Y_ADDR XBYTE[0x0100]
#define CMD XBYTE[0x0200]
#define DAT XBYTE[0x0300]

unsigned char code zk[32] = {//请
0x00, 0x47, 0x20, 0x23, 0x00, 0xEF, 0x20, 0x23, //left
0x22, 0x23, 0x22, 0x23, 0x2A, 0x32, 0x22, 0x02,
0x48, 0xFC, 0x40, 0xF8, 0x40, 0xFE, 0x08, 0xFC, //right
0x08, 0xF8, 0x08, 0xF8, 0x08, 0x08, 0x28, 0x10};

unsigned char code picture[];
main()
{
unsigned int x,y;
unsigned int i;
unsigned char j,k,z,m,n;

/////////////////////以下是清屏功能////////////////////////////////////
//用红色清屏
CMD = 0x04;DAT = 0x1f;//背景色
```

```
CMD = 0x08;//启动填充操作
for(y=0;y<250*10;y++);//延时 16.6 毫秒
CMD = 0x00;//退出填充操作
/////////////////////以上是清屏功能/////////////////////////////////////

////////8 点写模式下写入单色图片 80x80/////////////////////////////////
CMD = 0x02;DAT=0xe0;//前景色
CMD = 0x04;DAT=0xff;//背景色
CMD = 0x01;
for (y=0; y<80; y++)
        {
        X_ADDR = 0;
        X_ADDR = 0;
        Y_ADDR = y;
        for ( n=0; n<10; n++) DAT = picture[y*10+n];
        }
//////////////////8 点写模式下写汉字////////////////////////////////
CMD = 0x02;DAT=0xe0;//前景色
CMD = 0x04;DAT=0xff;//背景色
CMD = 0x01;
for ( n=0; n<=15; n++ )
{
        X_ADDR = 0;
        X_ADDR = 0;
        Y_ADDR = n;
        DAT = zk[n];
        DAT = zk[n+16];
 }

}
unsigned char code picture[800]={//，80x80
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xC0, 0x00, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x01, 0x80, 0x00, 0xC0, 0x03, 0xFF, 0x03, 0xFF, 0xC0,
0x00, 0x01, 0x80, 0x00, 0xC0, 0x06, 0x06, 0x00, 0x00, 0x00,
0x00, 0x7F, 0xFC, 0x01, 0x80, 0x06, 0x06, 0x00, 0x00, 0x00,
0x00, 0x63, 0x0C, 0x7F, 0xFF, 0x07, 0xFC, 0x00, 0x00, 0x00,
0x00, 0x63, 0x18, 0x01, 0x80, 0x0C, 0x0C, 0x3F, 0xFF, 0xC0,
0x00, 0xC6, 0x18, 0x03, 0x80, 0x0F, 0xFC, 0x00, 0x60, 0x00,
0x00, 0xC6, 0x30, 0x03, 0x80, 0x03, 0x60, 0x07, 0xD8, 0x00,
0x00, 0xFF, 0xF0, 0x07, 0x80, 0x36, 0xCC, 0x06, 0xD8, 0x00,
0x00, 0x0C, 0x00, 0x0D, 0x80, 0x36, 0xDC, 0x0C, 0xCC, 0x00,
0x00, 0x0C, 0x00, 0x0C, 0xC0, 0x1E, 0xF0, 0x39, 0x86, 0x00,
0x00, 0x0C, 0x00, 0x38, 0xE0, 0x1D, 0xE0, 0x61, 0x86, 0x00,
0x00, 0x18, 0x00, 0x60, 0x70, 0x0D, 0xC0, 0xC1, 0x80, 0x00,
0x00, 0x18, 0x01, 0xC0, 0x31, 0xFF, 0xFC, 0x07, 0x00, 0x00,
0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x03, 0xCF, 0xFE, 0x38, 0x20, 0x47, 0xB8, 0x00, 0x00,
0x00, 0x01, 0x87, 0x32, 0x60, 0x60, 0xCC, 0xD8, 0x00, 0x00,
0x00, 0x00, 0xC4, 0x30, 0xC0, 0xE1, 0xCC, 0xD8, 0x00, 0x00,
0x00, 0x00, 0xC4, 0x31, 0xF0, 0xE1, 0xCE, 0x9E, 0x00, 0x00,
0x00, 0x00, 0x68, 0x31, 0x99, 0x62, 0xC7, 0x9B, 0x00, 0x00,
0x00, 0x00, 0x68, 0x31, 0x9A, 0x64, 0xCD, 0xDB, 0x00, 0x00,
0x00, 0x00, 0x30, 0x31, 0x9B, 0xF7, 0xEC, 0xDB, 0x00, 0x00,
0x00, 0x00, 0x30, 0x31, 0x98, 0x60, 0xCC, 0xDB, 0x00, 0x00,

```
0x00, 0x00, 0x30, 0x78, 0xF0, 0x60, 0xC7, 0x96, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
```

以上原程序可到 www.yuanfengkeji.com 上下载。

14 售后支持说明

我们将尽最大努力保证所有售出的产品符合质量检测要求。

液晶模块属于元器件类产品。如果在使用过程中液晶显示器出现了故障，我们将为您提供下列服务：

1、显示器自购买之日起计算保修时间，保修期限为一年。

2、由于显示器本身质量问题所导致液晶模块工作不正常的，我公司将提供免费的维修服务，必要时可以更换显示器。

3、由于用户使用不当(例如：静电，焊接、连线不当，过流、过压使用)等原因导致该显示器受损的，我公司将尽力维修；但将酌情收取相应的维修成本或

更换元器件的费用。

4、因为液晶屏的物理损伤所造成液晶显示器不能正常工作的，该模块一般只能作报废处理。

5、在我公司购买的液晶显示器出现故障需要返修的时，请尽量详细描述该显示器的故障现象，以便我公司技术人员判断故障原因，更好的为您提供维修服务。

15 运输损坏处理办法

1、为确保您收到的货物准确无损，请在送货人在场的情况下打开包装检验货物。

2、如果您收到的货物因运输不当而受损，您有权向运输公司要求索赔。或原封不动地保留货物、货物包装材料及附件，并及时与我公司联系。

16 注意事项

使用前请认真阅读下列注意事项，避免不必要的损坏！

1、　确保是在电源关闭的状态下拔插线缆。

2、　液晶屏为玻璃易碎制品；任何跌落、敲打和强烈振动都可能导致玻璃破裂；禁止用力按压液晶屏的显示区域；安装时不能用力挤压液晶屏幕和边框；应注意液晶屏的整体平整度，避免外力导致液晶屏 “弯曲"、“扭曲”

3、　由于液晶屏的可视特性会因视角的不同而有所改变，所以在装配时，应该充分考虑使用者的合适视角，调整出液晶屏最佳的视角位置。

4、　在使用或储存中，应小心液晶屏的表面偏光片，避免被坚硬物体划伤。（严禁将坚硬物品置于液晶屏之上）

5、　显示器驱动电路中，逆变器部分带有高压，操作时请避免接触，避免造成不必要的损伤。

6、　液晶屏在低于规定的温度、温度范围下工作、存储，可能造成液晶结晶，而导致不可恢复的损坏；如果在高于规定的温度、温度范围下工作、存储，液晶可能变成各向同性的液体从而无法恢复液晶态。请在本显示器允许温度范围内保存和使用（详见性能指标栏中提供的参数）

7、　在液晶显示器背面的电路驱动部分有电子元器件，在设计、装配、处置过程中应避免挤压、损坏，否则可能因损坏电子元器件，而造成液晶屏的功

能障碍。

8、 本产品为精密仪器，禁止自行打开、拆卸或改装。否则可能会损坏液晶屏，而造成无法正常使用。

9、 使用和保存中应保持液晶屏表面的清洁，如若水滴长时间滞留在液晶屏上，可能会导致液晶屏表面变色或出现污斑；故当屏幕表面有污迹时，需使用纯棉或软质布擦拭；严禁使用汽油、酒精或其他化学药剂擦拭液晶屏幕。

10、 在使用完毕或发生故障时请及时关闭本机电源。

11、 为防止火灾或电击危险，请勿将本机放置在淋雨或潮湿的地方。

12、 如屏幕破损，液晶遗漏在手上或衣服上时请用肥皂和清水洗净即可。

13、 液晶模块采用了 CMOS-LSIs，所以处理液晶模块时，一定要有良好的防静电保护和有效的接地措施。

14、 为不断提升性能，本显示器和说明书会做进不断的升级和改版 ，恕不另行通知。

感谢使用郑州远峰电子定做生产的嵌入式彩色液晶显示器

郑州远峰电子

郑州市中州商场 **3** 楼北厅 **3429** 号

电话：**0371-68582512**

传真：**0371-68582512**

为了提高产品的整机性能和可靠性，可能会对产品的硬件和软件做一些小的调整，恕不另行通知，如果产品的实际情况与说明书有某些不一致，以实物为准。

对于说明书内容有不同理解的，以本公司技术部门解释为准。